CLAIM LISTING:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A distributed processing system comprising:

a plurality of processing objects; and

an object management system,

wherein at least two of the processing objects comprise an interface in the object management system defining a deferred procedure call from first processing object to a second processing object according to an interface definition language, the interface definition language including a source code instruction having a syntax including an "interface" keyword, an interface name, a return value type, a function name, at least one function argument, and an identifier from a set of values that includes an identifier of a one-way procedure call, an identifier of a two-way blocking procedure call, and an identifier of a deferred procedure call.

2. (Original) The distributed processing system of claim 1, the distributed processing system further comprising:

a control plane comprising at least one processing object; and

a data plane comprising at least one processing object,

wherein the interface in the object management system defines a deferred procedure call from a processing object in the control plane to a processing object in the data plane.

3. (Currently Amended) The distributed processing system of claim 2, wherein the processing object in the data plane comprises an interface to one or more microengine processes.

4. (Original) The distributed processing system of claim 1, the distributed processing system further comprising:

a control plane comprising at least one processing object; and

a data plane comprising at least one processing object,

wherein the interface in the object management system defines a non-blocking procedure call from a processing object in the control plane to a processing object in the data plane.

5. (Currently Amended) The distributed processing system of claim 4, wherein the processing object in the data plane comprises an interface to one or more microengine processes.

6. (Original) The distributed processing system of claim 1, wherein the first processing object comprises:

logic to execute a crosscall stub to initiate the procedure call to the second processing object; and

logic to execute a callback skeleton in response to receipt of a return value from the second processing object,

wherein the crosscall stub and callback skeleton are derived from a compilation of a deferred procedure call instruction formatted according to the interface description language.

7. (Original) The distributed processing system of claim 6, wherein the second processing object comprises logic to execute a crosscall skeleton in response to a procedure call from the crosscall stub.

8. (Original) The distributed processing system of claim 7, wherein the crosscall stub and callback skeleton comprise image generated from a compilation of a single procedure call interface definition formatted according to the interface description language, and wherein the second processing object comprises logic to asynchronously call back the first processing object in response to the procedure call.

9. (Currently Amended) A processing system comprising:

a first processing core adapted to process information in data packets received from a transmission medium; and

a second processing core comprising:

a plurality of processing objects, at least one processing object having an interface with one or more processes hosted on the first processing core; and

an object management system, wherein at least two of the processing objects comprise an interface in the object management system defining a deferred procedure call from first processing object to a second processing object according to an interface definition language, the interface definition language including a source code instruction having a syntax including an "interface" keyword, an interface name, a return value type, a function name, at least one function argument, and an identifier from a set of values that includes an identifier of a one-way procedure call, an identifier of a two-way blocking procedure call, and an identifier of a deferred procedure call.

10. (Original) The processing system of claim 9, wherein the first processing core comprises a plurality of microengine processing elements.

11. (Original) The processing system of claim 9, the processing system further comprising:

a control plane comprising at least one processing object; and

a data plane comprising at least one processing object,

wherein the interface in the object management system defines a deferred procedure call from a processing object in the control plane to a processing object in the data plane.

12. (Currently Amended) The processing system of claim 11, wherein the processing object in the data plane comprises an interface to one or more microengine processes.

13. (Original) The processing system of claim 9, the distributed processing system further comprising:

a control plane comprising at least one processing object; and

a data plane comprising at least one processing object,

wherein the interface in the object management system defines a non-blocking procedure call from a processing object in the control plane to a processing object in the data plane.

14. (Currently Amended) The processing system of claim 13, wherein the processing object in the data plane comprises an interface to one or more microengine processes.

15. (Original) The processing system of claim 9, wherein the first processing object comprises:

logic to execute a crosscall stub to initiate the procedure call to the second processing object; and

logic to execute a callback skeleton in response to receipt of a return value from the second processing object,

wherein the crosscall stub and callback skeleton are derived from a compilation of the deferred procedure call instruction formatted according to the interface description language.

16. (Original) The processing system of claim 15, wherein the second processing object comprises logic to execute a crosscall skeleton in response to a procedure call from the crosscall stub.

17. (Original) The processing system of claim 15, wherein the crosscall stub and callback skeleton comprise image generated from a compilation of a single procedure call interface definition formatted according to the interface description language, and wherein the second processing object comprises logic to asynchronously call back the first processing object in response to the procedure call.

18. (Currently Amended) A method comprising:

~~formatting~~ accessing a deferred procedure call instruction in a source code module corresponding with a first processing object, the deferred procedure call instruction being formatted according to an interface description language, the instruction having a syntax including an "interface" keyword, an interface name, a return value type, a function name, at least one function argument, and an identifier from a set of values that includes an identifier of a one-way procedure call, an identifier of a two-way blocking procedure call, and an identifier of a deferred procedure call; and

compiling the source code module to provide a crosscall stub image and a callback skeleton image based upon the deferred procedure call instruction, wherein the callback skeleton image comprises instructions enabling execution of the first processing object following a procedure call to a second processing object and prior to receipt of a return value at the callback skeleton.

19. (Original) The method of claim 18, wherein the method further comprises linking the crosscall stub image and callback skeleton image with a crosscall skeleton image associated with the second processing object.

20. (Original) The method of claim 19, wherein the first processing object comprises a processing object in a control plane of a distributed processing system and the second processing object comprises a processing object in a data plane of the distributed processing system.

21. (Original) The method of claim 19, wherein the first processing object comprises a processing object in a data plane of a distributed processing system and the second processing object comprises a processing object in a control plane of the distributed processing system.

22. (Currently Amended) An article comprising:

a storage medium comprising machine-readable instructions stored thereon for:

~~formatting~~ accessing a deferred procedure call instruction in a source code module corresponding with a first processing object, the deferred procedure call instruction being formatted according to an interface description language, the instruction having a syntax including an "interface" keyword, an interface name, a return value type, a function name, at least one function argument, and an identifier from a set of values that includes an identifier of a one-way procedure call, an identifier of a two-way blocking procedure call, and an identifier of a deferred procedure call; and

compiling the source code module to provide a crosscall stub image and a callback skeleton image based upon the deferred procedure call instruction, wherein the callback skeleton image comprises instructions enabling execution of the first processing object following a procedure call to a second processing object and prior to receipt of a return value at the callback skeleton.

23. (Original) The article of claim 22, wherein the storage medium further comprises machine-readable instructions stored thereon for linking the crosscall stub image and callback skeleton image with a crosscall skeleton image associated with the second processing object.

24. (Original) The article of claim 23, wherein the first processing object comprises a processing object in a control plane of a distributed processing system and the second processing object comprises a processing object in a data plane of the distributed processing system.

25. (Original) The article of claim 23, wherein the first processing object comprises a processing object in a data plane of a distributed processing system and the second processing object comprises a processing object in a control plane of the distributed processing system.

26. (New). The method of claim 18, wherein the syntax comprises a syntax of:

interface <name> {oneway|twoway|deferred} <type> <function_name> (<arguments>).

27. (New). The article of claim 22, wherein the syntax comprises a syntax of:

interface <name> {oneway|twoway|deferred} <type> <function_name> (<arguments>).